Continuous Tensor Field Approximation of Diffusion Tensor MRI data

Sinisa Pajevic[1], Akram Aldroubi[2], and Peter J. Basser[3]

[1] National Institutes of Health, MSCL/CIT; Bethesda, MD, USA
[2] Dept. of Mathematics, Vanderbilt University; Nashville, TN, USA
[3] National Institutes of Health, STBB/NICHD; Bethesda, MD, USA

Abstract. Diffusion Tensor MRI (DT-MRI) measurements are a discrete noisy sample of an underlying macroscopic effective diffusion tensor field, $\underline{\mathbf{D}}(\mathbf{x})$, of water. This field is presumed to be piecewise continuous/smooth at a gross anatomical length scale. Here we describe mathematical framework for obtaining an estimate of this tensor field from the measured DT-MRI data using a spline-based continuous approximation. This methodology facilitates calculation of new structural quantities and provides a framework for applying differential geometric methods to DT-MRI data. A B-spline approximation has already been used to improve robustness of DT-MRI fiber tractography. Here we propose a piecewise continuous approximation based on Non-Uniform Rational B-Splines (NURBS), which addresses some of the shortcomings of the previous implementation.

1 Introduction

Diffusion tensor MRI provides a measurement of an effective diffusion tensor of water, $\underline{\mathbf{D}}^{\text{eff}}$, in each voxel within an imaging volume [1]. These diffusion measurements are inherently discrete, noisy and voxel-averaged. Here we treat DT-MRI data as discrete noisy samples of an underlying macroscopic piecewise continuous diffusion tensor field, $\underline{\mathbf{D}}(\mathbf{x})$, where, $\mathbf{x} = (x, y, z)$ are the spatial coordinates in the laboratory frame of reference. This field is presumed to be piecewise continuous or smooth at a gross anatomical length scale, an assumption based on the known anatomy of many soft fibrous tissues, including white matter, muscles, ligaments, and tendons. One of our objectives is to develop a mathematical framework to estimate this piecewise continuous field, $\underline{\mathbf{D}}(\mathbf{x})$, from discrete noisy DT-MRI measurements. A reliable estimate of this field enables us to use differential geometric methods directly. Additionaly, it enables computation and display of intrinsic architectural or microstructural MRI features based upon tissue fiber geometry [2,3]. Some previously suggested characteristics are curvature and torsion of the individual fiber tracts, as well as the properties of the tangent field, e.g. twisting, bending, and diverging [4]. Here we focus on estimating curvature or the fiber tracts (tangent field) but also show the architectural features of the tensor field itself. Estimating such quantities accurately using measured diffusion tensor data and interpolation is difficult, since their evaluation requires spatial differentiation of noisy tensor quantities. Below we show that they can be calculated more reliably and robustly using continuous tensor field approximation.

Originally, estimating tensor field from sample tensor data was performed using B-spline approximation [5]. It was used with DT-MRI data to elucidate fiber tract trajectories, which can be done by integrating the fiber direction (vector) field [6]. Integrating a noisy direction vector field can result in fiber trajectories that wander off course. Using a smoothed representation of the direction field, obtained from the continuous representation of $\underline{\mathbf{D}}(\mathbf{x})$, however, can improve the fidelity of tract following [6]. Establishing connectivity and continuity of neural pathways can also benefit from the development of this specialized tensor field processing methodology. These tasks require establishing continuous links between different regions of the brain, or assessing disjunctions between them. Finally, there are a number of generic image processing tasks one would like to perform on high dimensional DT-MRI data, since no signal processing framework currently exists for these. These include: filtering noise, sharpening edges, detecting boundaries; compressing, storing and transmitting large image files; interpolating and extrapolating tensor data; resampling data at different resolutions (e.g., rebinning); extracting textural features, segmenting images, clustering data, and classifying tissues; and detecting statistical outliers. The B-spline approximation provides the mathematical underpinnings for performing these tasks both rapidly and efficiently [7]. However, the problem with it is that it introduces smoothing in the data uniformly and isotropically and is incapable of dealing with discontinuities. The smaller structures as well as sudden or rapid changes (edges, high curvatures, etc.) will be distorted at the levels of approximation/smoothing required to alleviate the noise effects. To achieve a more efficient approximation we use Non-Uniform Rational B-Splines (NURBS). They allow for discontinuities and can describe complex piecewise continuous geometrical shapes with many fewer parameters than the original B-spline approximation.

Although there are other approaches for finding an approximate tensor field [8–10], in this paper we focus on a mathematical framework for continuous approximation based on splines. We describe and compare two different methods for computing approximated tensor field: (i) the previously proposed method using B-splines and (ii) a new method that uses NURBS. We compare this two and discuss future work.

2 Continuous Approximation and Representation of Discrete Tensor Data

To construct a continuous approximation to a diffusion tensor field, we start with a set of basis functions (approximants) whose linear combinations define an approximation space. In [5], to make the approximation scheme practicable, we required it possess the following properties: (P1) The set of basis functions must be sufficiently rich to represent the diffusion tensor field precisely and accurately; (P2) The mathematical description of the approximation space is computationally tractable; (P3) The approximation of the diffusion tensor field is implemented using algorithms that are fast, robust, and accurate. In this paper we also require (P4) the approximation scheme must be able to produce a piecewise continuous representation. We will see later that this can be done using NURBS, which will provide even richer set of basis functions (strengthening P1), however, the requirement for speed in P3 will have to be relaxed.

To meet these requirements in general, we use atomic spaces [13], which are a generalization of shift invariant spaces. In particular, we choose an atomic space, $S_A(\mathbf{x}, \underline{B})$, such that any function in that space, $\underline{T}(\mathbf{x})$ is of the form

$$\underline{T}(\mathbf{x}) = \sum_{i=1}^{N_r} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} P^m(i,j,k) \times \underline{B}^m(\mathbf{x}, \mathbf{Q}_{i,j,k}) \tag{1}$$

In other words, each approximant in the approximation space, $\underline{T}(\mathbf{x})$, is a weighted sum of a finite number of tensor field generators, $\underline{B}^m(\mathbf{x}, \mathbf{Q}_{i,j,k}),\ m = 1, \ldots, N_r$. The $P^m(i,j,k)$ are the coefficients for the total of $N_r N_x N_y N_z$ basis functions and are the first set of parameters of the approximation model. The other parameters that describe the basis functions are lumped into $\mathbf{Q}_{i,j,k}$, and can be different for different basis functions as indicated.

We showed previously that finding the tensor field generator could be reduced to finding a continuous representation of each of its individual tensor components [7]. To represent the field of the symmetric diffusion tensor, we proposed the following six orthogonal tensor-field generators used in Eq. 1 to define the tensor approximation space :

$$\begin{aligned}
\underline{B}^1(\mathbf{x}) &= b^1(\mathbf{x}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad & \underline{B}^2(\mathbf{x}) &= b^2(\mathbf{x}) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\underline{B}^3(\mathbf{x}) &= b^3(\mathbf{x}) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad & \underline{B}^4(\mathbf{x}) &= b^4(\mathbf{x}) \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\underline{B}^5(\mathbf{x}) &= b^5(\mathbf{x}) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad & \underline{B}^6(\mathbf{x}) &= b^6(\mathbf{x}) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}
\end{aligned} \tag{2}$$

Each tensor field generator $\underline{B}^m(x)$, can now be expressed in terms of a single function, $b^m(\mathbf{x})$, which now serves as a basis for the i^{th} component of the tensor field. Based on the choice for this function we distinguish between two implementations for the field generators; the original one that used B-splines [5] and the new one that uses NURBS.

3 B-spline approximation

With the B-splines we choose $b^m(\mathbf{x})$ to be a product of one-dimensional functions, i.e., $b^m(\mathbf{x}) = f^m(x)g^m(y)h^m(z)$. The basis functions are now separable in two ways, first with respect to the components of the tensor, and second with the respect to the coordinates. Finding the continuous field $\underline{\mathbf{D}}(\mathbf{x})$ can be reduced to applying a one dimensional approximation algorithm along x, y and z coordinates sequentially within the imaging volume for each component [7].

The $f^m(x), g^m(y), h^m(z)$ are B-spline functions[14,15] which are obtained by repeated convolutions of the simple box function (Fig. 1a) The number of convolutions determines the order of the B-spline, i.e., linear, quadratic, cubic, etc. The use of the separable basis function provides also an easy way to account for the nonuniform resolutions in x, y, and z directions in some DT-MRI acquisitions.

Using B-spline functions has several advantages: B1) the generators have finite spatial extent (i.e., finite support), which speeds up and simplifies digital processing algorithms; B2) they can be evaluated fast in recursive fashion (as well as analytically, faster) in terms of splines of lower degree; B3) the derivatives of B-splines can be expressed recursively in terms of the original B-splines; B4) by changing the polynomial order or degree of the B-spline functions, we can control the degree of smoothness and differentiability of our continuous approximation; B5) by adjusting the (scale) parameters of the B-spline representation we can choose between interpolation (fitting data points exactly) and approximation (fitting data points approximately); B6) invariant representation under affine as well as perspective transformations; B7) possess the convex-hull property; and B8) B-spline functions naturally generate multi-resolution structures that are useful in analyzing signals and images at different length scales.

Additionally the separable multi-dimensional spline functions behave well for the cubic and higher order splines as demonstraed in Figure 1b. It shows that the two-dimensional spline function $b^m(\mathbf{x})$ constructed as a product of linear one-dimensional B-splines is anisotropic (i.e., shows preferential directions) and will produce artifacts when used for scaling (i.e., smoothing) a general tensor field. However, when the cubic B-splines are used these artifacts are negligibly small and the $b^m(\mathbf{x})$ constructed in this way perform nearly as well as the true two-dimensional isotropic basis functions, but are much more computationally efficient to implement. In our applicatation we use mainly the cubic B-splines. If higher order derivatives are needed, it is advisable to use B-splines of higher polynomial order than three to preserve the isotropic properties of the multidimensional basis functions.

Another advantage of using B-splines is that they need a very few additional parameters. In fact the simpliest implementation $\mathbf{Q}_{i,j,k}$ can consist of only one scale parameter, Δ, but typically we use three additional scale parameters, Δ_x, Δ_y, and Δ_z, which control the degree of smoothness along

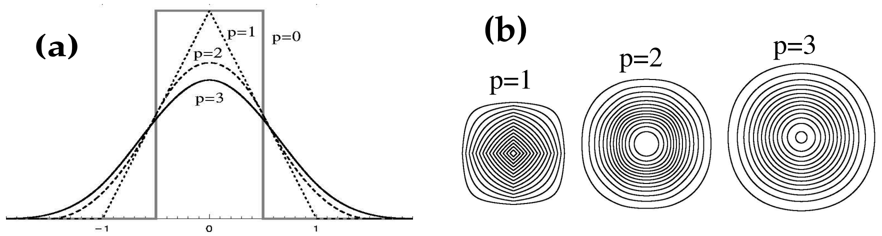

Fig. 1. (a) 1-D B-spline functions of degree p=0 through 3. The B-spline of degree n is obtained by n-fold convolutions of the box function (p=0) as indicated. **(b)** 2-D separable B-spline basis functions with degrees p=1, 2, and 3

each direction; the shifts on a uniform grid within the imaging volume, are indicated by k, l, and m. The generator for B-splines in this case is written as

$$\underline{B}^m(\mathbf{x}, \mathbf{Q}_{i,j,k}) = \underline{B}^m(\mathbf{x}, \boldsymbol{\Delta}) = \underline{B}^m(x\Delta_x - i, y\Delta_y - j, z\Delta_z - k) \tag{3}$$

The optimal choice of the coefficients, $P^m(i, j, k)$, for a given choice of the B-spline and scale parameters is the one that minimizes the least-squared difference between the original tensor data and the approximated diffusion tensor field [7].

3.1 Implementing B-spline approximation

Once the polynomial order of the B-spline is chosen (generally cubic), then for a given DT-MRI data set, we calculate B-spline coefficients in the x, y, and z-directions for each of the six independent diffusion tensor elements using the spatial separability property described above. Thus, we perform 18 1-D transforms on the tensor data set. The 1-D B-spline approximation we use is based on a scale conversion algorithm which finds the optimal approximation to the original signal at a given scale Δ[14,15,12]. The only difference in our implementation is the exclusion of the postfiltering step, as described in the block diagram of the algorithm in [16]. The task of this algorithm is to find the minimal least square approximation of the original signal in the space scaled down by factor Δ. This algorithm efficiently obtains the B-spline coefficients by projecting the B-spline expansion of the original signal onto the scaled space basis. This algorithm is not exact, i.e., it does not provide a mathematically precise projection between the two spaces. However, the deviations from the exact solution are mostly formal in nature. In practice, the performance of this algorithm is nearly optimal, while gaining in speed and efficiency. This and other details of our implementation are described in the Appendix of [5].

An important step in the implementation is also to choose the appropriate scale parameters. When the scale parameters equal 1, the continuous representation becomes interpolation; when one or more of the $\varDelta_i$ is less than 1, the continuous representation becomes a data reduction technique that approximates or fits the discrete tensor data. Since the $\varDelta$ is the ratio of the number of unknown parameters to the number of measured data points for the 1-D approximation the scale parameters can only take on specific rational values, $\{\varDelta\}_N$, which designates the rational number closest to $\varDelta$ that contains N in the denominator. For DT-MR images N is usually large enough to allow sufficient precision in the range of the scale parameter values between 0 and 1. We reduce the number of scale parameters by choosing only one $\varDelta$ and by assigning the three values of the model as $\{\varDelta\}_{N_x}, \{\varDelta\}_{N_y}, \{V_r\varDelta\}_{N_z}$, where V_r is the voxel aspect ratio ($V_r \approx 2$) in our case), thus making the grid of B-spline coefficients more uniform. Ideally, the value of $\varDelta$ should be twice the ratio of the maximal spatial frequency of the pure (noise-free) signal and the sampling frequency. Note, however, that our approximation method is not a simple low-pass filter and the projecting onto a smoother space is *not* the same as smoothing. The first one provides the least square fit while the latter does not, in general. In cases where structures within the image appear at all length scales, the choice of $\varDelta$ is empirical as the structures on the small scales (single or a few voxels) must be blurred in order to improve estimates of large structures of the diffusion tensor field elsewhere.

The B-spline approximation, although successfully applied for the fiber tracking application[6] (see Figure 5), does not provide a reliable framework for applying general methods of differential geometry to DT-MRI data. Calculations of curvature, torsion and other differential geometric quantities were highly unreliable[5].

4 Non-Uniform Rational B-Splines (NURBS)

Non-Uniform Rational B-Splines, or NURBS as they are widely known, are a powerful tool to describe and model complex curves and surfaces using a small number of parameters[17]. It is a much richer set of basis functions which generalize many concepts of ordinary B-splines. To introduce the NURBS model we focus on the 1-D model, since here too the multidimensional models are derived using products of basis functions.

There are three main groups of parameters that describe a NURBS model. The first is the knot vector, U, which controls the non-uniformity along a particular dimension, the second is a set of weights, $\mathbf{W}$, one for each basis functions, and the third are the "control points", which correspond to the B-spline coefficients of our model, but we adopted this commonly used jargon.

The knot vector is a set of monotonically nondecreasing numbers in the real interval $[a, b]$ which parametrizes a given curve, i.e.,

$$U = \{\underbrace{a, ..., a}_{p+1}, u_{p+1}, ..., u_{i-1}, u_i, u_{i+1}, ..., u_{m-p-1}, \underbrace{b, ..., b}_{p+1}\} \tag{4}$$

As can be seen some values can be repeated and an important quantity is the multiplicity of the knot, n_m, which is equal to the number of times a given value is repeated. In this way it is possible to control the continuity of the curve. Each basis function is C^k-continuous, $k = p - n_m$ at a knot with multiplicity n_m, and C^∞ continuous elsewhere. The $p + 1$ repetition of the end points of the interval is just a statement that at the edges the curve is discontinuous. Alternatively, one can lower the multiplicity at the end points by using boundary conditions.

The set of basis functions is obtained using the recursive structure of the B-splines except that now the factors in front of the interacting B-splines of lower order are not constant but are functions of the knot vector. They are called Non-Uniform B-Splines (NUBS). The NUBS basis functions are obtained using the following recursion:

$$B_{i,p}(u) = \left(\frac{u - u_i}{u_{i+p} - u_i}\right) B_{i,p-1}(u) - \left(\frac{u - u_{i+p+1}}{u_{i+p+1} - u_{i+1}}\right) B_{i+1,p-1}(u) \tag{5}$$

$$B_{i,0}(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$

and the k^{th} derivate at point u can be obtained using

$$B^{(k)}_{i,p}(u) = p\left(\frac{B^{(k-1)}_{i,p-1}(u)}{u_{i+p} - u_i} - \frac{B^{(k-1)}_{i+1,p-1}(u)}{u_{i+p+1} - u_{i+1}}\right) \tag{6}$$

Figure 2a shows a set of NUBS for the given values of spline degree and the knot vector. Once a set of NUBS is obtained we use the second set of parameters, $\mathbf{W}$, which are the weights associated with each of the basis functions. The rational basis functions $R_{i,p}(u)$ corresponding to the i^{th} NUB basis function, $B_{j,p}$, is obtained as

$$R_{i,p}(u) = \frac{B_{i,p}(u) w_i}{\sum_{j=0}^{n} B_{j,p}(u) w_j} \tag{7}$$

The third set of parameters are the coefficients P^m of the spline model, or in NURBS parlance, *control points.* The control points are actually tuplets of B-spline coefficients grouped together to reflect the geometry of the model. For example, for a 3-D space curve the B-spline coefficients of the $x(u)$,

$y(u)$ and $z(u)$ functions are grouped together to form a 3-D control point, $\mathbf{P}_i = (P_x^i, P_y^i, P_z^i)$. The NURBS curve model $\mathbf{C}(u)$ can now be written as

$$\mathbf{C}(u) = \sum_{i=0}^{n} R_{i,p}(u)\mathbf{P}_i \tag{8}$$

We see that it is essentially a linear model. Thus here too we can choose between interpolation (number of control points same as the number of data points) or the least square fit (fewer control points). This model can be extended using the function products, as described in section 2, to surfaces, volumes and ultimately to tensor fields. Here we finally write our NURBS tensor model in terms of *control tensors*, $\mathbf{D}^c_{i,j,k}$,

$$\mathbf{D}(x,y,z) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} R_{i,j,k}(x,y,z)\mathbf{D}^c_{i,j,k} \tag{9}$$

where $R_{i,j,k}(x,y,z)$ is a new 3-D rational function defined from NUBS, which now require three different knots vectors U, V, and S and can be of different degree in each dimension (p,q,r), i.e.,

$$R_{i,j,k}(x,y,z) = \frac{B_{i,p}(x)B_{j,q}(y)B_{k,r}(z)w_{i,j,k}}{\displaystyle\sum_{i'=0}^{n}\sum_{j'=0}^{m}\sum_{k'=0}^{l} B_{i',p}(x)B_{j',q}(y)B_{k',r}(z)w_{i',j',k'}} \tag{10}$$

Eq. 9 is equivalent to Eq. 1, except that here the N_r independent components of the tensor field are lumped together into control tensor. This geometric interpretation can be very useful. For example, since the convex hull property has much tighter bounds in the case of the NURBS model we can use positive semidefiniteness of the control tensors to enforce the same property for the tensor field at any point in the space.

The NURBS model shares all the good properties of the B-spline model (B1-B8) but has important additional advantages: (N1) precisely represents a large family of mathematical curves, piecewise polynomials, conic sections (circles, ellipses, hyperbolas, parabolas), Bezier curves, and very efficiently arbitrary shapes (N2) can control the degree of smoothness and continuity, including discontinuous functions, thus suitable for piecewise continuous representation.

However, although they enable fast evaluation in a recursive fashion at any point in the space (like B-spline), obtaining the appropriate parameters of the model is much harder now. We assume here that one wants to use the properties N1-N2, otherwise NURBS model can be simplified. After all, the B-splines are a special case of NURBS with uniformly spaced knot vector and all weights equal. Here we mainly refer to NURBS as model that requires non-uniform knots and varying weights.

In the case of one dimensional data one can still obtain useful NURBS fits relatively quickly. Figure 3a shows a fit to synthetic 2-D data (noisy samples of two different 2-D curves were joined together to create an apparent discontinuity). The solid line indicates the fit to a NURBS model and one can see that even with a discontinuity, the NURBS describes the curve very well with only 14 control points. The B-spline model was incapable of describing such curve.

The image inset in Figure 3b shows a 2D-projection of a tract (solid yellow line) onto a slice of DT-MRI volume with color coded orientations. The tract passes through the corpus callosum and spans from one end of the brain to the other. Figure 3b shows side by side the same tract in 3-D described with NURBS model (left) and B-spline model (right). The B-spline model, parameterized with B-spline coefficients needs a total of 3*130=390 parameters to produce as faithful representation of the space curve as the NURBS model did with only 15 control points (total of 60 parameters). Such a sparser parameter space enables more efficient explorations of connectivity, and more importantly can significantly alleviate the noise effects. In the next section we compare B-spline and NURBS models on important example of estimating local curvature of the fiber tract.

5 B-spline vs NURBS Comparison on Curvature Estimation

Estimating curvature is problematic for noisy data. The curvature of a fiber tract, or a space curve, is defined as

$$\kappa(u) = \left| \frac{d\mathbf{t}(u)}{du} \right| = \frac{|\dot{\mathbf{r}} \times \ddot{\mathbf{r}}|}{|\dot{\mathbf{r}}|^3} \tag{11}$$

Since higher derivatives are involved, this estimate is very sensitive to noise (each derivation acts as a linear ramp high pass filter). Another look at the problem is that the fiber tracts are not polynomial and their estimates are noisy. Thus in order to faithfully depict a given curve one has to represent curves with a relatively high number of B-spline coefficients, thus sampling the noise often. The curvature of noise is infinite and thus the local estimates of the radius will be biased towards zero, besides being also very noisy. We reported previously that the B-spline approximation did not produce satisfactory results in this regard. Here we compare B-spline estimates with NURBS estimates.

We test curvature estimation on a simulated space curve consisting of four circular arcs with radii R_c=100,10,5,35 in arbitrary units. We then sampled 50 points with sampling error of 1%, which are indicated as solid black circles on the inset in upper right corner of Figure 5. The inset also shows the 15 control points of the fit, together with the “ideal” control points (light blue circles) which could describe such curve exactly. There are total of 10 "ideal" points but not all are shown since in many cases they are very close to the control points obtained from the fit. We see that the NURBS fitting routine can be improved further, however, even this imperfect fit provides significant improvement over the B-spline approximation. The solid black line in Figure 5a indicates the true radius, with the exception of the three inflection points where the curvature is infinite. The solid blue line indicates the NURBS estimates outperforms the B-spline approximation for any level of smoothness, as described in the caption. The spike in the NURBS estimate occurs at the inflection point where the radius of curvature is infinite.

In figure 6 we determined the radius of curvature for every voxel in one of the slices using both NURBS estimate and the B-spline approximation with $\Delta = 0.2$ (since such choice produced relatively stable estimates in figure 5, however with significant loss in resolution. This loss of resolution is evident in figure 6a, where it appears that very little variation of the radius of curvature is occuring, for example, in the splenium of the corpus callosum. The NURBS estimate in 6b does show significant variation. Even though the lower curvature structures appear "spotty" in the NURBS estimate they indicate errors of only 20 % (a good precision for the curvature radius).

6 Discussion and Conclusion

The continuous approximation methodology takes noisy, voxel-averaged, and discrete statistical samples of an underlying macroscopic effective diffusion tensor field as its input, and produces a piecewise continuous, smooth tensor field approximation as its output. Besides being able to recover the original noiseless tensor field reliably, the approximation schemes substantially reduces bias of the mean and variance of various quantities derived from the tensor field, e.g., Trace($\underline{\mathbf{D}}(\mathbf{x})$). This new methodology also facilitates following nerve and other fiber tract trajectories *in vivo*. New MR features or parameters that characterize structure, architecture, and functional assessment of tissues can be developed from this continuous representation of DT-MRI data. As analytical functions are used to approximate the diffusion tensor field, we can evaluate and display quantities such as the gradient tensors[5], which cannot be evaluated accurately from noisy DT-MRI data.

6.1 Microscopic field (underlying) vs macroscopic field (voxel averaged)

The microscopic tensor field is one that describes water diffusion on a microscopic scale, whereas the macroscopic effective tensor field describes the tensor field on a voxel scale. If we assume no intercompartmental mixing of spins, the measured macroscopic tensor field is just the voxel-average of this microscopic tensor field. While these macro and micro fields should be

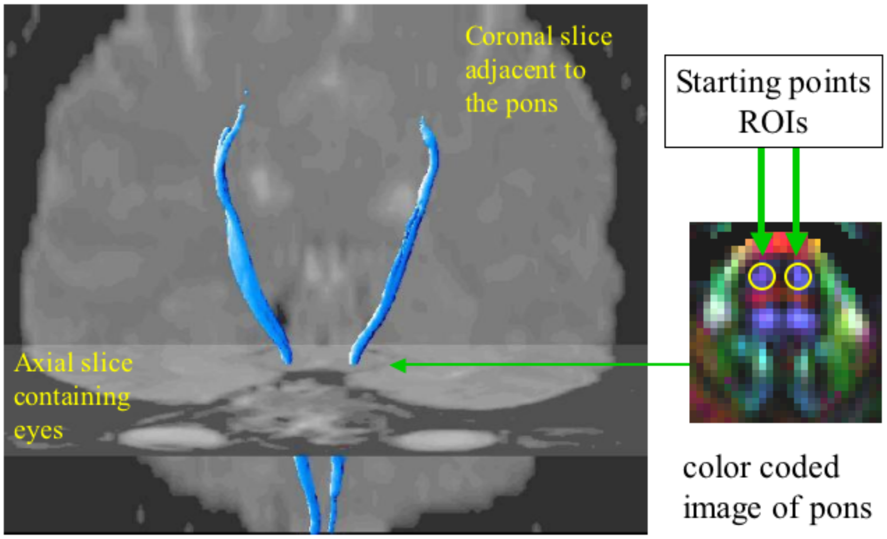
Coronal slice adjacent to the pons
Axial slice containing eyes
Starting points ROIs
color coded image of pons

Fig. 2. Fiber tracts result from integrating along the tangent direction of the B-spline approximated tensor field, and with starting points choosen from the two circular regions in the area of pons. The obtained result agrees well with known anatomical data.

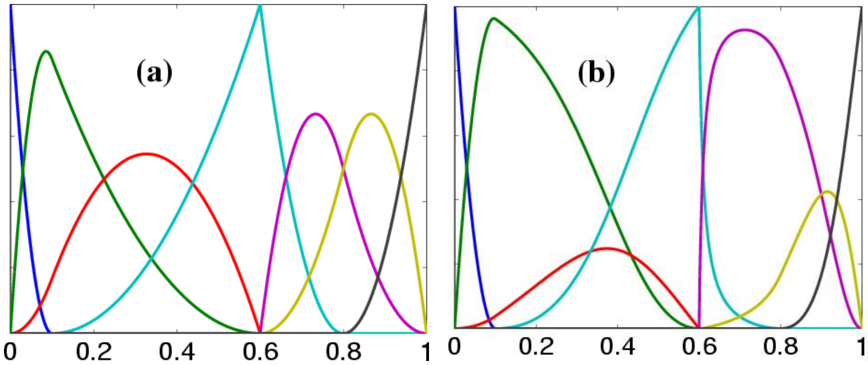

Fig. 3. **(a)** A set of 1-D NUB-basis functions with $p = 2$ and the knot vector $U = [0\ 0\ 0\ 0.1\ 0.6\ 0.6\ 0.8\ 1\ 1\ 1]$. **(b)** A set of rational basis functions (NURBS) obtained using the NUBs in (a) and by changing the weights for the $3^{rd}, 4^{th}$ and 5^{th} basis function to 0.2, 0.5 and 5, respectively. The remaining NUBs had weights $w = 1$.

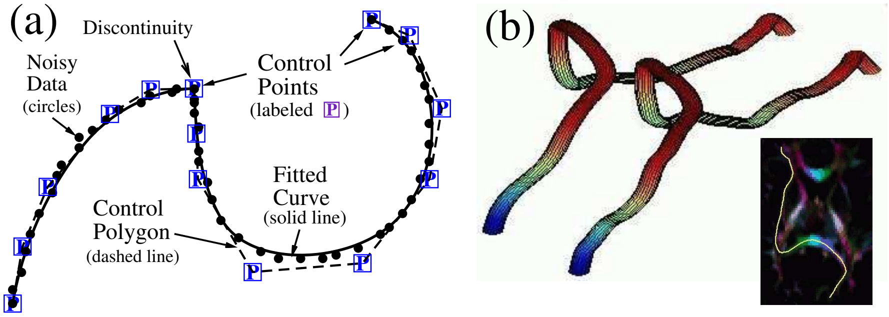
(a)
Discontinuity
Noisy Data (circles)
Control Points (labeled P)
Fitted Curve (solid line)
Control Polygon (dashed line)
P
(b)

Fig. 4. **(a)** 2-D curve NURBS model fit to noisy data **(b)** 3-D curve NURBS model fit to fiber tracking data (indicated on the inset image).

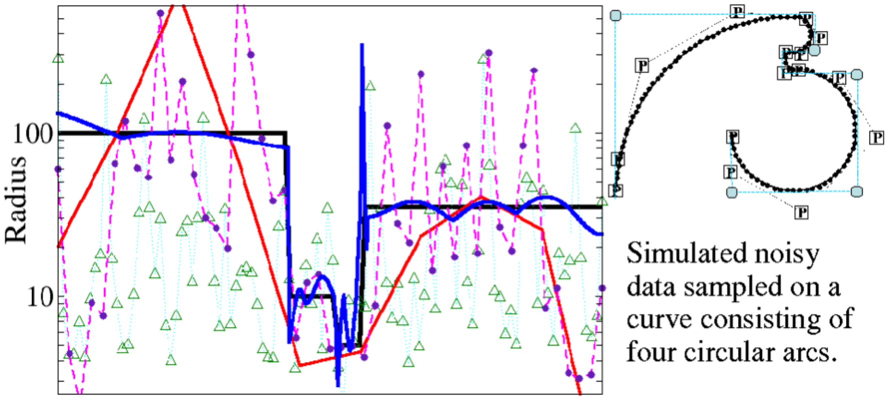

Fig. 5. Radii of curvature obtained from a noisy data set of points sampled from a curve consisting of four circular arcs (see the inset in upper right corner) with radii 100,10,5,35. The sampling error was 1%. The solid black line indicates the true radius (at inflection points the raidus is infinite). The solid blue line indicates the NURBS fit, while B-spline approximation estimates are labeled as follows: $\Delta = 1$, i.e., interpolation (green triangles), $\Delta = 0.5$ (purple dots), or $\Delta = 0.2$ (red solid line). Note that the original curve could have been described with only 10 control points (the light blue circles, not all shown)

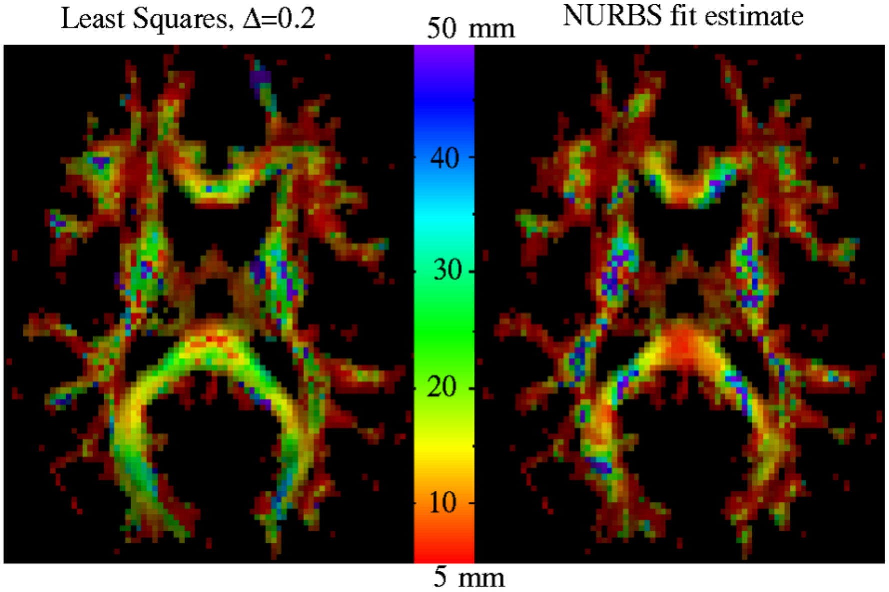

Fig. 6. Color coded images of the radius of curvature obtained at each voxel of the given slice using B-spline approximation with $\Delta = 0.2$ (left) and NURBS (right), with colorbar indicating the scales. We see that the NURBS estimates are capable of showing the spatial variation of the fiber curvature.

similar in regions containing tissue whose distribution of fiber direction is uniform within the voxel, in tissues whose distribution of fiber direction is non-uniform, such as regions where fibers diverge or converge (splay), bend or twist, branch or merge, a significant disparity may exist between these fields. Generally, in these regions, the macroscopic field with be a powder average of the heterogeneous microscopic tensor field within a voxel. An important long-term goals is to develop techniques to identify regions in which such powder averaging occurs, and then attempt to infer the microscopic tensor field from macroscopic voxel-scale measurements there using additional information from other sources[18].

6.2 NURBS vs B-spline

The B-spline approximation is not nearly as efficient as the NURBS model. Its estimation of quantities involving higher derivatives is inaccurate and thus model is not adequate for applying differential geometric methods. Another shortcoming of the B-spline approximation method is that it forces continuity of the tensor field at boundaries or interfaces, where there is no physical requirement to impose continuity. This results in high approximation errors at the edges of the structures.

NURBS can account for piecewise continuity by modifying knot vector. NURBS produce promising results for 1-D curves, and to some extent with surfaces. However, to date, we have been unable to obtain an efficient tensor model, mainly due to an extremely large parameter space and a the very rich model, which is difficult to fit (many local minima). It appears that the NURBS methodology will have to be used differently than the B-spline approximation, and a long computation will be required to obtain the model, using various stochastic fitting methods (simulated annealing, genetic algorithms, etc.) Once the model is obtained it will be possible to explore its geometry, tract fibers, and run various differential geometric models with almost the same efficiency as with B-splines. Work is underway to improve NURBS fitting and at the same time to generalize this continuous tensor field approximation to treat internal boundaries and discontinuities in the tensor field more naturally and robustly by using control tensors that do not use the actual coordinates x, y, and z for parameterization. In other words the control tensor will contain information about its position in space. The tensor model is now written as

$$\mathbf{D}(u,v,s) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} R_{i,j,k}\mathbf{D}^{c}_{i,j,k}(\mathbf{r}_{i,j,k}) \qquad (12)$$

Effectively, the control tensor is now a 9-dimensional quantity (6 independent tensor components plus 3 spatial dimensions).

References

1. P.J. Basser, J. Mattiello, and D. Le Bihan. MR diffusion tensor spectroscopy and imaging. Biophys. J. 66(1), 259-67 (1994).
2. P.J. Basser. Inferring microstructural features and the physiological state of tissues from diffusion-weighted images. NMR Biomed 8(7-8), 333-44 (1995).
3. P.J. Basser, and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. J Magn Reson B 111(3), 209-19 (1996).
4. P.J. Basser. New histological and physiological stains derived from diffusion-tensor MR images. Ann N Y Acad Sci 820, 123-38 (1997).
5. S. Pajevic, A. Aldroubi and P.J. Basser. A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. J Magn Reson, 154, 85-100 (2002).
6. P.J. Basser, S. Pajevic, C. Pierpaoli, A. Aldroubi, and J. Duda. In Vivo Fiber-Tractography in Human Brain Using Diffusion Tensor MRI (DT-MRI) Data. Magn. Reson. Med., 44:625-632 (2000).
7. A. Aldroubi, and P.J. Basser, Reconstruction of vector and tensor fields from sampled discrete data. in ”Contemporary Mathematics”. L.W. Baggett, D.R. Larson, editors) pp. 1-15, AMS, Providence, RI (1999).
8. Poupon, C., Clark, C.A., Frouin, F., Regis, J., Bloch, I., Bihan, D.L., Bloch, I., Mangin, J.-F., Regularization of diffusion-based direction maps for the tracking of brain white matter fascicles. NeuroImage 12, 184-195 (2000).
9. J.-F. Mangin, C. Poupon, C. Clark, D. Le Bihan and I. Bloch, Eddy-Current Distortion Correction and Robust Tensor Estimation for MR-Diffusion Imaging,
10. C.F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.A., Jolesz, R. Kikinis, Processing and visualization for diffusion tensor MRI. Med Image Anal, 6, 93-108 (2002). Lecture Notes in Computer Science, 2208, 186 (2001).
11. P.J. Basser, S. Pajevic, C. Pierpaoli, A. Aldroubi, and J. Duda. Fiber-Tractography in Human Brain Using Diffusion Tensor MRI (DT-MRI); 2000; Denver. p 784.
12. A. Aldroubi, M. Eden, and M. Unser. Discrete spline filters for multiresolutions and Wavelets of l2. SIAM Math. Anal. 25, 1412-1432 (1994).
13. A. Aldroubi. Oblique projections in atomic spaces. Proceedings of the Amer. Math. Soc. 124, 2051-2060 (1996).
14. M. Unser, A. Aldroubi, and M. Eden. B-Spline Signal Processing: Part II-Efficient Design and Implementation. IEEE Transactions on Signal Processing 41(2), 834-848 (1993).
15. M. Unser, A. Aldroubi, and M. Eden. B-Spline Signal Processing: Part I-Theory. IEEE Transactions on Signal Processing 41(2), 821-833 (1993).
16. M. Unser, A. Aldroubi, and M. Eden. Enlargment or reduction of digital images with minimum loss of information, IEEE Transactions on Image Processing 4, 247-258 (1995).
17. L. Piegl, W. Tiller, The NURBS Book, Springer-Verlag, 1997
18. D.S. Tuch, R.M. Weiskoff, J.W. Belliveau, and V.J. Wedeen. High Angular Resolution Diffusion Imaging of the Human Brain; 1999; Philadelphia. p 321.